

	Direction Générale des Études Technologiques Institut Supérieur des Études Technologiques de Djerba Département Technologies de l'Informatique	
Matière : Programmation Web 2	<b>Épreuve : Devoir de synthèse</b>	
Classe : L1-TI1-S2	Durée : <b>1h 30mn</b> Nombre de page : <b>2</b>	
Enseignants : Aymen BOUREGBA, Olfa HAMROUNI JEMLI, Mouna KORT SOUISSI, Ratiba BEN AZZOUN		

### **Exercice 1 (6 points)**

On veut encoder un arbre généalogique dans un format XML. Un arbre généalogique présente les ascendants et les descendants d'une personne. Une personne a un attribut facultatif « id » de type ID et un attribut obligatoire « genre » pouvant prendre au choix les valeurs « h » et « f ». Une personne est identifiée par un « nom », un « âge », un « groupe\_sanguin » et a un « père ».

L'élément « père » peut être vide et possède un attribut obligatoire « ref », de type IDREF (identifiant de la personne qui représente le père).

L'élément « groupe\_sanguin » possède un attribut obligatoire « famille » qui prend la valeur « A, B, O ou AB » et un attribut « type » qui prend la valeur « négatif » ou « positif ».

1. Écrire le document XML décrivant la généalogie suivante: Maher a un père nommé Salah, son âge est 23 ans et son groupe sanguin est AB négatif. Salah a un père nommé Sami, son âge 50 ans et son groupe sanguin est A positif. Sami de 75 ans a un groupe sanguin B négatif.
2. Écrivez une DTD interne permettant de décrire cet arbre

### **Exercice 2 (5 points)**

Écrire le code de la DTD permettant de spécifier ce qui suit :

1. Un élément « paiement » étant soit un chèque, soit une carte bancaire soit du cash.
2. Un élément « projet » contenant un ensemble de « personnes » (2 personnes) et un ensemble de « tâches » (au moins une).
3. Un élément « personne » identifié par un nom, un âge et une adresse (plusieurs adresses ou aucune).
4. Un attribut « type » pour l'élément « Animal » qui est vide. L'attribut est obligatoire et est formé d'un seul mot.
5. Une entité paramètre nommé « info » permettant de remplacer (titre | éditeur) dans le code suivant : <!ELEMENT document ((titre | éditeur)\*, nom).

### **Exercice 3 (9 points)**

Soit le fichier « PC.xml » suivant.

```
<?xml version="1.0" encoding="UTF-8"?>
<PC xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="PC.xsd">
  <logiciel>
    <systeme>windows 7.0</systeme>
    <systeme>ubuntu 14.01</systeme>
    <application type="bureautique" lang="ar">Open Office</application>
    <application type="comptabilite">Express Accounts</application>
  </logiciel>
  <materiel>
    <processeur nbits="64">
      <vitesse>3.2Ghz</vitesse>
    </processeur>
    <memoire capacite="8" unite="Go" />
    <disque>
      <taille>150To</taille>
      <pilote>disk.inf</pilote>
    </disque>
  </materiel>
</PC>
```

Écrire un schéma pour ce fichier, « PC.xsd », en tenant compte des informations suivantes.

- Tous les attributs sont obligatoires excepté l'attribut « lang ».
- L'attribut « nbits » ne peut avoir que deux valeurs possibles : 32 ou 64.
- L'élément « pilote » est un nom de fichier composé de lettres et chiffres, ayant toujours l'extension « .inf ».
- L'élément « disque » peut apparaître au moins une fois et au maximum 3 fois.

***Bon travail...***

## CORRECTION

### CORRECTION ex1

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE arbre [
  <!ELEMENT arbre (personne*)>
    <!ELEMENT personne (nom, parent?, age, grouSang)>
    <!ATTLIST personne id ID #IMPLIED
      genre (h | f) #REQUIRED>
    <!ELEMENT parent EMPTY>
    <!ATTLIST parent ref IDREF #REQUIRED>
    <!ELEMENT nom (#PCDATA)>
    <!ELEMENT age (#PCDATA)>
    <!ELEMENT groupSang EMPTY>
    <!ATTLIST groupSang famille (A|B|AB|O) #REQUIRED type (negatif | positif) #REQUIRED>
  ]>
<arbre>
  <personne id="M" genre="h">
    <nom>Maher</nom>
    <parent ref="S"/>
    <age>23</age>
    <groupSang type="negatif">AB</groupSang>
  </personne>
  <personne id="S" genre="h">
    <nom>Salah</nom>
    <parent ref="Sh"/>
    <age>50</age>
    <groupSang type="positif">A</groupSang>
  </personne>
  <personne id="Sh" genre="h">
    <nom>Sami</nom>
    <age>75</age>
    <groupSang type="negatif">B</groupSang>
  </personne>
</arbre>
```

### Correction ex2

1. <!ELEMENT paiement (cheque|CB|cash)>
2. <!ELEMENT projet(personne ?, tache ?)>
3. <!ELEMENT personne(nom, age, adresse\*)>
4. <!ELEMENT animal empty>  
  
<!ATTLIST animal type NMTOKEN #REQUIRED>
5. <!ENTITY % info "titre | editeur">  
  
<!ELEMENT document ((%info),nom)>

### Correction ex3

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="PC" type="TPC" />

  <xs:complexType name="TPC">
    <xs:sequence>
      <xs:element name="logiciel" type="Tlogiciel" />
      <xs:element name="materiel" type="Tmateriel" />
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="Tlogiciel">
    <xs:sequence>
      <xs:element name="systeme" type="xs:string" maxOccurs="unbounded" />
      <xs:element name="application" type="Tapplication" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="Tmateriel">
    <xs:sequence>
      <xs:element name="processeur" type="Tprocesseur" />
      <xs:element name="memoire" type="Tmemoire" />
      <xs:element name="disque" type="Tdisque" maxOccurs="3" />
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="Tapplication">
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="type" type="xs:string" use="required" />
        <xs:attribute name="lang" type="xs:string" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>

  <xs:complexType name="Tprocesseur">
    <xs:sequence>
      <xs:element name="vitesse" type="xs:string" />
    </xs:sequence>
    <xs:attribute name="nbits">
      <xs:simpleType>
        <xs:restriction base="xs:integer">
          <xs:pattern value="32|64" />
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:complexType>

  <xs:complexType name="Tmemoire">
    <xs:attribute name="capacite" type="xs:string" use="required" />
    <xs:attribute name="unite" type="xs:string" use="required" />
  </xs:complexType>
```

```
<xs:complexType name="Tdisque">
  <xs:sequence>
    <xs:element name="taille" type="xs:string" />
    <xs:element name="pilote">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:pattern value="[a-zA-Z0-9]+.(inf)"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
</xs:schema>
```